A Practical Evaluation in Openstack Live Migration of VMs using 10Gb/s Interfaces

Md Israfil Biswas, Gerard Parr, Sally McClean, Philip Morrow and Bryan Scotney
School of Computing and Information Engineering
Ulster University, Coleraine, Northern Ireland
Email: {mi.biswas, gp.parr, si.mcclean, pj.morrow, bw.scotney}@ulster.ac.uk

***Abstract*—Live Migration (LM) of Virtual Machine (VM) is a process of transferring a working VM from on host to another host of a different physical machine without interfering the VM. In datacentre networks, LM enables flexibility in resource optimisation, fault tolerance and load balancing. However, in real time, the resource consumption and latency of live VM migration reduce these benefits to much less than their potential. In this paper, we present the results of an experimental study that evaluates LM in our unique high speed optical fibre network connecting Northern Ireland, Dublin and Halifax (Canada). We observe that using Pre-Copy LM extra large amounts of stressed memory leads to non convergence over high latency paths. However, using Post-copy LM the total migration time as well as downtime is dominated by specific memory utilisation patterns inside the virtualised guest. We experience variation in total migration time and downtime using Post-Copy LM considering Quality of Service (QoS) parameters, which can have significant impact in the cloud applications performance.**

Keywords—Virtual Machine, Live Migration, Openstack.

I. INTRODUCTION

Resource scheduling and management frameworks became absolute necessary for continuous management and maintenance of datacentres where LM is a key feature [1]. LM is a core function of rapidly evolving technology Virtualisation. While virtualisation provides a range of benefits to computing systems such as improved resource utilisation, management, application isolation, portability and system reliability, LM replace running VMs seamlessly across distinct physical hosts.

Recently, VM live migration technology has attracted considerable interest for datacentre management and cluster computing. There were also many other studies on the migration strategy available for a variety of application cases concerning the issues of live VM migration. However, few studies are available on the issue of efficient migration over high latency paths. In [2] Bobroff et al showed that low-latency migration could reduce resource requirements up to 50% and service-level agreement violations by up to 20%, and they demonstrated the correlation between resource efficiency and migration latency. In our previous work [3] we also showed that prioritising tasks for the nearest servers or with low latency not only improve the QoS but also demonstrates better utilisation of the resources. Hence, for the purpose of, all the VMs hosted by 10Gb/s interface would be potential candidates for migration. However, different migration latency may lead to significant differences in performance. Considering total migration time, downtime etc previous studies demonstrated that it could vary significantly between different workloads, ranging from milliseconds to tens of seconds in the case of high latency. This is mostly due to the diversity of VM configurations and workload characteristics. For instance, the initial memory size of a VM and applications' memory access pattern are critical factors that have a decisive effect on the migration latency, i.e. the total time a VM is undergoing performance penalty and high power state.

In this paper we have evaluated the performance of live VM migration using 10Gb/s interfaces where our network infrastructure is based on Openstack nova development [4]. We also have used shared storage system 'Network File System (NFS)' for this work of virtualisation.

The rest of the paper is organised as follows: Section II describes the pre-copy and post-copy live VM migration techniques that has been evaluated in this work with discussion to related works, Section III describes our cloud testbed under high speed fibre optic 10Gb/s network infrastructure, Section IV describes the experiment setup and configuration for this work, Section V presents the performance evaluation of our Cloud testbed using 10Gb/s interfaces in deferent latency paths with experiments and results. Finally, the paper concludes with the Conclusions with a view for future work.

II. LIVE MIGRATION OF VIRTUAL MACHINES

A VM instance contains its states, memory and emulated devices, which is transferred from one hypervisor to another with no possible downtime during live migration. This leads to the two characteristics that have been investigated in this paper:

- *Total Migration Time*: the preparation time from start of the LM process until the virtualisation framework notifies that the source host can be deactivated.
- *Downtime*: the phase during migration when there is user service unavailability or the execution of VM is stopped.

This section illustrates the pre-copy and the post-copy of LM approach to understand the advantages and disadvantages.

Pre-Copy Live Migration: Memory is transferred before VM allocation in the Pre-Copy approach of live VM migration.

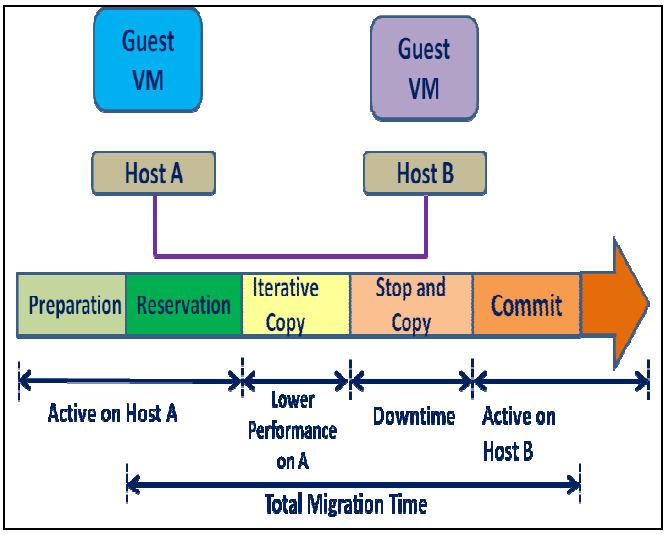

Fig. 1. Pre-Copy Live Migration Process

However, the issue is how to copy memory while it is re-dirtied over and over again by the guest VM? This is solved by first copying all the memory followed by intervals of copying newly dirtied pages until the remaining state is small enough as shown in Fig.1. Hence, the total migration time in this process is the Reservation time, Iterative Pre-Copy time (this could be several rounds depending on the dirtied pages), the time required to 'Stop and Copy' and the Commit time (i.e., the time that is running in the destination host). Pre-Copy LM is implemented by all most all hypervisors (e.g., Xen, Qemu, VMWare). Pre-Copy LM is often challenged fast memory dirtying applications.

Post-Copy Live Migration: Post-Copy LM is the process where the Transfer-Memory is moved only after VM relocation. this is important to ensure that VM performance is not degraded after the relocation for the network bound page faults. Hence, fast interconnects and improved page fault mechanisms are required to solve this issue, which is challenged by fast memory reading applications. The main advantage of Post-Copy is lower downtime. This is because CPU and short VM stats will be migrated while the VM is stopped as shown in Fig. 2. TABLE I shows the pros and cons between pre-copy and post-copy LM.

A. *Related Works*

Live migration of VMs using pre-copy were analysed by Clark et al in [5] and introduced the concept of a *writable working set (WWS)*. They found low downtimes for their workloads on dual core systems. For mostly stack pages they found the WWS is relatively small for commercial workloads and for scientific workloads it embraces most of the system memory.

Ibrahim et al [6] analysed the characteristics of iterative pre-copy live VM migration for memory intensive applications and proposes an optimised pre-copy strategy that dynamically adapts to the memory change rate in order to guarantee convergence. The algorithm is implemented in KVM, detects memory update patterns and terminates migration when no improvements in downtime.

Compared to pre-copy technique of bandwidth bound, post-copy is a latency bound technique. Hines et al [7] analysed post-copy LM to reduce the migration time to show promising results for commercial workloads. Post-copy techniques have not been thoroughly evaluated for scientific workloads particularly for remote machines that started and migrate without copying the memory pages but copied on demand. Moghaddam et al [8] analysed post-copy LM to reduce the downtime for copying changed memory pages that may significantly slowdown the migrated instance.

TABLE I. COMPARISON BETWEEN PRE-COPY AND POST-COPY LIVE MIGRATION

Issues	Pre-Copy LM	Post-Copy LM
Copy of Host VM's Memory	Easier Copy	Delayed Copy
Downtime	Longer and unpredictable(depending on writable working set)	Shorter
Total Migration Time	Shorter	Longer
After Migration	High Performance	Low Performance due to page falut
Network Bandwidth	No properly Utilisation	Effective Utilisation

Aidan Shribman et al analysed pre-copy and post-copy migration in [9], where they have proposed a page reordering policy 'Least Recently Used (LRU)' that has lower chance of re-dirtied and migrated earlier for pre-copy LM. They also propose delta encoder, Xor Binary Zero Run Length Encoding (XBZRLE) to reduce the cost of the page re-send. In post-copy LM they have proposed Remote Direct Memory Access (RDMA) for low-latency resolution of network-bound page faults and pre-paging or, pre-fetching to reduce the overall page faults integrating page faulty mechanism and hybrid LM.

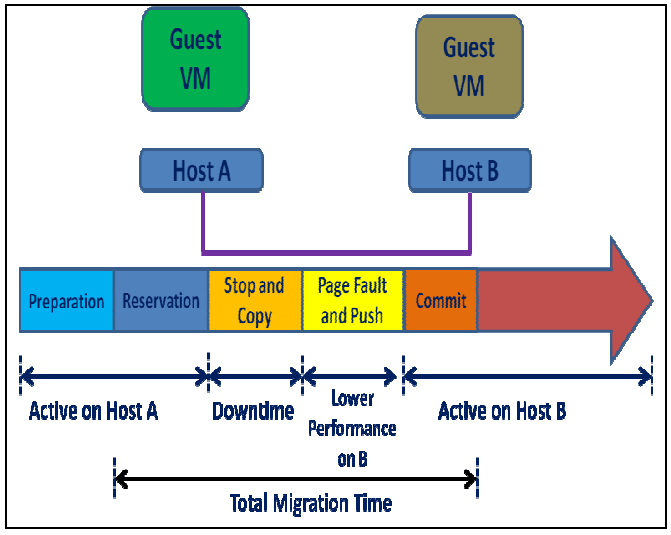

Fig. 2. Post-Copy Live Migration Process

In contrast, we provide a practical evaluation of live VM migrations in a high speed optical fibre network, where hosts are remotely connected with 10Gb/s interfaces. In this paper, we investigate the pre-copy and post-copy LM techniques that deal with 10Gb/s interference and represents comparative analysis of various strategies dealing with the effect of QoS parameters. This work can be helpful to the service provider, cloud service developers and cloud service consumers to identify the interference affecting the application performance.

III. ULSTER INTERCONNECTED TEST-BED

Ulster University Cloud testbed (UlsterCloud) is designed considering three intelligent controllers that can automate the management of resources in the provider network and in the cloud computing datacentres respectively. UlsterCloud aims to provide IaaS, PaaS, SaaS to local and remote users while securely linked with enterprise sites. The test-bed is designed primarily to provide a platform for development of next generation monitoring, intelligence and orchestration tools interface with next generation standard tools to provide seamless resource monitoring and orchestration with flexible network management.

The testbed incorporates a range of industry-standard servers, physical networking fabric and storage nodes to outperform existing virtualisation technologies at the server, router, and network levels to create dynamic resource pools that can be transparently connected to enterprises.

Our contract with Hibernia Atlantic [10] has provided fibre optic connection from Hibernia Cable Landing Stations (CLSs) and provided multipoint circuits between Coleraine (Northern Ireland), Dublin and Halifax (Canada) as shown in Fig. 3. The circuits from Hibernia Networks provide a direct 10Gb/s interface from Coleraine CLS to Ulster University campus. The other two hosts are connected to Hibernia Dublin CLS and Hibernia Halifax CLS. However, each can be incremented to the far-end CLS sites and able to cope with 10Gb/s and beyond 10Gb/s toward 100Gb for burst traffic at short intervals over our allocated wavelengths.

A. *The Testbed Build-out*

Our contract with Hibernia Network is to interconnect the three CLSs only. Hence, Hibernia is not providing any gateway to outside or, no Internet access to update the remote hosts. We are also not allowed to connect with the JANAET or, not allowed to bridge the two networks for security reason. As we need remote access to conduct software updates between the CLSs, we need secured outside connectivity.

Hence, an ADSL service for the Internet access is provided by a network Switch to the testbed at the Ulster University (shown in Fig. 4). Therefore, the switch connects the Hibernia CLSs with 10Gb/s interface and the ADSL link. The ADSL link is installed to operate at approximately 76Mb/s, only for server firmware updates and to comply with UK JISC rules.

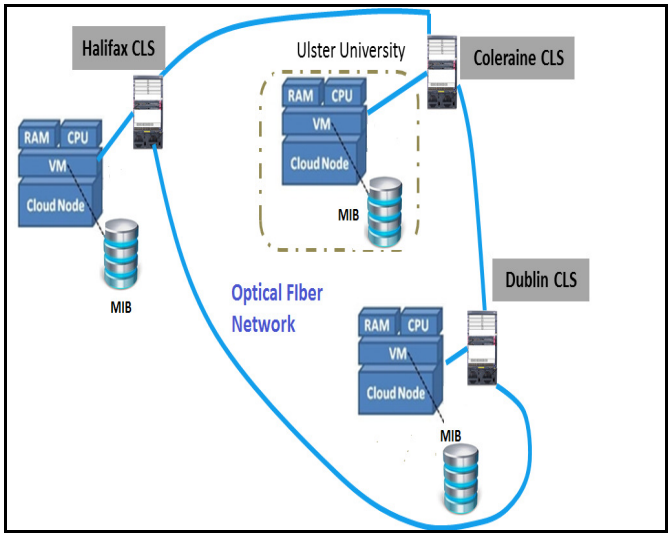

Fig. 3. Planned Ulster Interconnected testbed

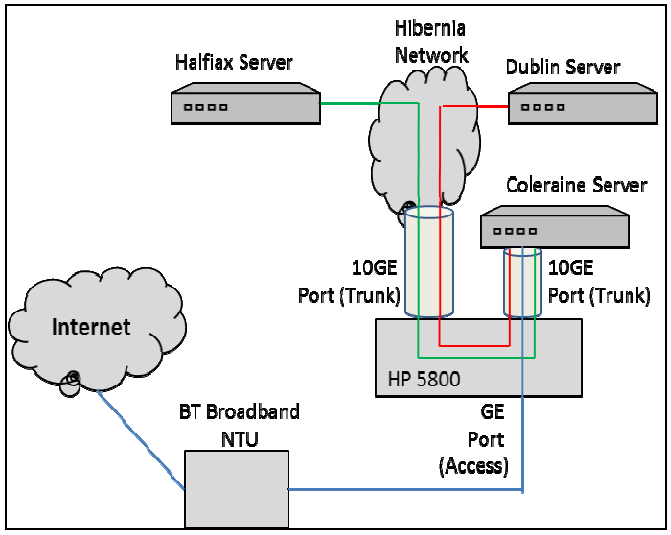

Fig. 4. Internet Access through ADSL connection (BT broadband) with HP switch 5800

B. *File Storage and Sharing through Virtualisation*

Network File System (NFS) Server for UlsterCloud is a Virtual Private Cloud (VPC) that provides seamless and secure virtualisation with file storage facility as illustrated in Fig. 5. The vision is efficient pooling of geographically secluded datacentre resources with optimised support for LM.

By default, migration only transfers in-memory state of a running domain for example memory, CPU state etc. Disk images are not transferred during migration but they need to be accessible at the same path from both hosts. Therefore, some kind of shared storage needs to be setup and mounted at the same place on both hosts like NFS. We have configured the NFS server at the Ulster University with the following capability and edited the *etc/exports* file to serve as a shared storage:

 # mkdir -p /exports/images
*/exports/images *(rw,no_root_squash)*

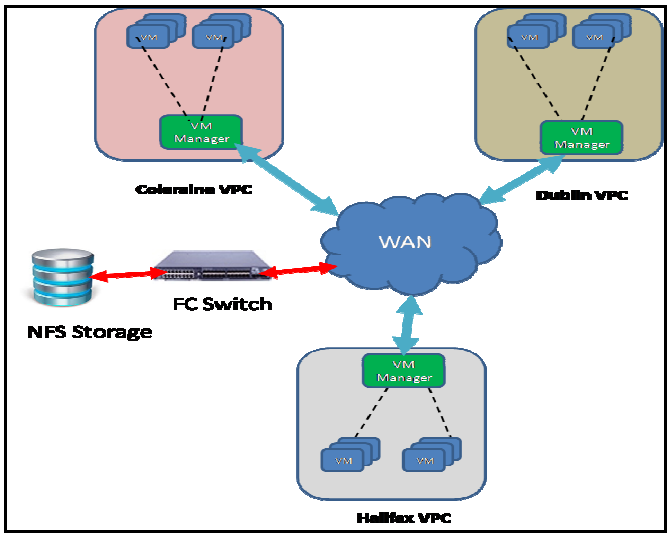

Fig. 5. UlsterCloud NFS Shared storage for VMs

The exported directory needs to be mounted at a common place of all the hosts that running libvirt, configuring the *etc/fstab* file:

IPaddress:/exports/images /vm_images nfs auto 0 0
 # mount /vm_/images

We observed that naive solution of exporting a local directory from one host using NFS and mounting it at the same path on the other host did not work. The directory used for storing disk images has to be mounted from shared storage on both hosts. Otherwise, the domain may lose access to its disk images during migration because source libvirtd may change the owner, permissions, and SELinux labels on the disk images once it successfully migrates the domain to its destination. Libvirt avoids doing such things if it detects that the disk images are mounted from a shared storage.

IV. System Configuration and Setup

In this experiment, we have used three physical machines Dell PowerEdge R815 (AMD Opteron 6366HE@ 3,6GHz, 128GB RAM) as a modules of a Blade server providing 10Gb/s network interfaces through its backplane. We installed Openstack with QEMU / libvirt with post-copy support. Two servers were configured purely as a compute node running nova and nova-network services. The third one was configured as both a compute node and the controller node providing also all the other management services.

Following system configuration is used for these tests:

- 3 nodes: 1 control node (Coleraine), 2 compute nodes (Dublin, Halifax)
- Openstack Icehouse release
- Nova 2.18.1
- QEMU 1.2.1
- Libvirt 0.10.2

A. *System setup*

The following configuration has been done for the experiment, some verification and check lists are omitted for the simplicity.

1. Network configuration
All hosts hypervisors are running in the same network/subnet.

1.1. DNS configuration
DNS configuration and consistency of */etc/hosts* file across all hosts are done.

1.2. Firewall configuration
The */etc/sysconfig/iptables* file is configured to allow libvirt listen on TCP port 16509 and added a record accepting KVM communication on TCP port within the range from 49152 to 49261.

```
-A INPUT -p tcp -m multiport --ports 16509 -m comment --comment "libvirt" -j ACCEPT
-A INPUT -p tcp -m multiport --ports 49152:49216 -m comment --comment "migration" -j ACCEPT
```

2. Libvirt configuration
We have configured the */etc/sysconfig/libvirtd* file of libvirt to enable the flag.

The */etc/libvirt/libvirtd.conf* file is configured to make the hypervisor listen TCP communication with none authentication. SSH keys for authentication are strongly recommended as authentication is set to NONE.

```
listen_tls = 0
listen_tcp = 1
auth_tcp = “none”
```

3. Nova configuration
To enable real live VM migration, we have set up *live_migration* flag in */etc/nova/nova.conf* file as openstack does not use real live VM migration mechanism as a default setting. This is because there is no guarantee that the migration is successful (e.g., faster dirtied pages than transferred to destination host).

```
live_migration_flag=VIR_MIGRATE_UNDEFINE_SOURCE,VIR_MIGRATE_PEER2PEER,VIR_MIGRATE_LIVE
```

B. *Live Migration Execution*

Firstly, we have checked the list available for VMs, and then checked VM details to determine which host an instance running on. After that we used commands to list the available compute hosts and to choose the host we want to migrate the instance to as this is very much secured and efficient in nova. Then migration of the instance is done to a new host.

Finally, we have checked the VM details and also confirmed if this has been done successfully.

V. EVALUATIONS

For the evaluation of live VM migration, we focused on measuring the followings:

- Migration duration or, the total migration time
- Duration of VM unavailability or, the VM downtime
- Amount of data transferred through the migration interface using stress tool [11]

All the experiments of real throughput of 10Gb/s interface is conducted using iperf [12] network bandwidth measurement tool. The configuration is using full 10Gb/s of local links and also limiting 10Gb/s for higher latency links (e.g., Coleraine-Halifax and Coleraine-Dublin links). Our observation in real throughput of 10Gb/s interface is 9.4 Gb/s between local servers as no restriction is implied. However, we achieved throughput up to 7 Gb/s between Coleraine-Dublin link with an average latency of 5.5ms and up to 3Gb/s between Coleraine-Halifax link with an average latency of 52.5ms.

TABLE II. POST-COPY LM TIME (SEC) WITH STRESS MEMORY

Stress [MB]	0	1024	2048	3078	4096	5120	6144	7168	8192
Coleraine-Local	7.2	10.9	14.9	18.2	21.8	27.1	30.2	34.9	41.1
Coleraine-Dublin	11.2	28.4	47.5	64.7	81.2	96.1	117.2	132.9	151.2
Coleraine-Halifax	11.3	28.5	47.6	65.0	81.4	98.6	118.4	133.4	152.8

We found very limited possibilities of using standard pre-copy algorithm for migration of live VMs with memory intensive applications between Coleraine-Dublin and Coleraine-Halifax links. Fig. 6 shows that using the pre-copy algorithm Coleraine-Halifax link is able to transfer only light or thin stressed memory of VMs. This is because, higher amounts of stressed memory leads to non convergence (e.g., the Coleraine-Halifax link). Using the fibre optic network in Coleraine-Dublin link, the pre-copy approach is able to transfer up to 50% higher size of the stress memory compared to the Coleraine-Halifax link. However, we have compared the same experiment with local 10Gb/s interface, where we observed that the migration using pre-copy approach can increases up to 400MB in approximately 15 sec.

Hence, the threshold of the amount of stressed memory is increased and thus extends possibilities of using the standard pre-copy algorithm for more use cases. Beyond the threshold value we observed the pre-copy approach fails also using the local link. We found the downtime varied from 300ms to 400ms in all cases when live migration converges.

The post-copy converges regardless of the amount of memory stressed and the network throughput. TABLE II and Fig.7 show linear increase of migration time with increasing amount of stressed memory in the Coleraine-Dublin and Coleraine-Halifax links.

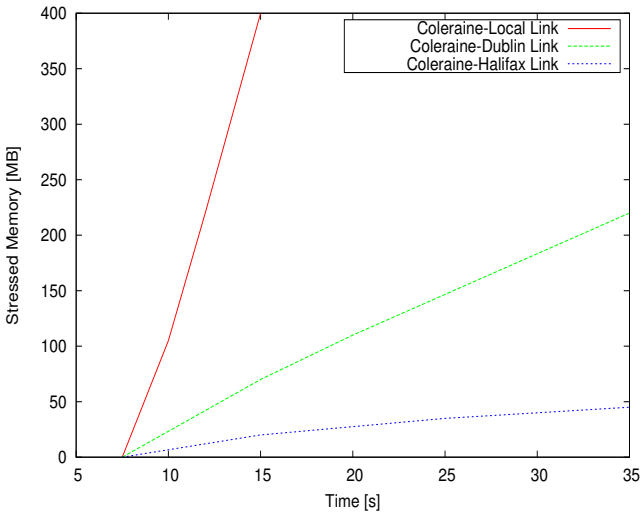

Fig. 6. Pre-Copy LM –Stressed Memory with Time (s)

However, in all the cases using 10Gb/s local connections leads to lower migration times. Migrating essentially light or thin VM, the total migration time can increase up to 56% in high latency path. For instance, migration time is 11.3s using Coleraine-Halifax link and 11.2s using Coleraine-Dublin link whereas, 7.2s is calculated in 10Gb/s local link. As the size of stressed memory increases, the total migration duration for both links significantly varies compared to the migration duration of local VMs. For instance, extra large 8 GB of VM's took 152.8s in Coleraine-Halifax and 151.2s in Coleraine-Dublin link to migrate but the same memory using 10Gb/s local link takes only 41.1s to transfer.

TABLE III. POST-COPY DOWNTIME (SEC) WITH STRESS MEMORY

Stress [MB]	0	1024	2048	3078	4096	5120	6144	7168	8192
Coleraine-Local	0.2	0.2	0.3	0.4	0.4	0.4	0.5	0.4	0.5
Coleraine-Dublin	0.3	0.3	0.4	0.4	0.5	0.5	0.5	0.6	0.6
Coleraine-Halifax	0.4	0.5	0.5	0.5	0.6	0.6	0.6	0.7	0.7

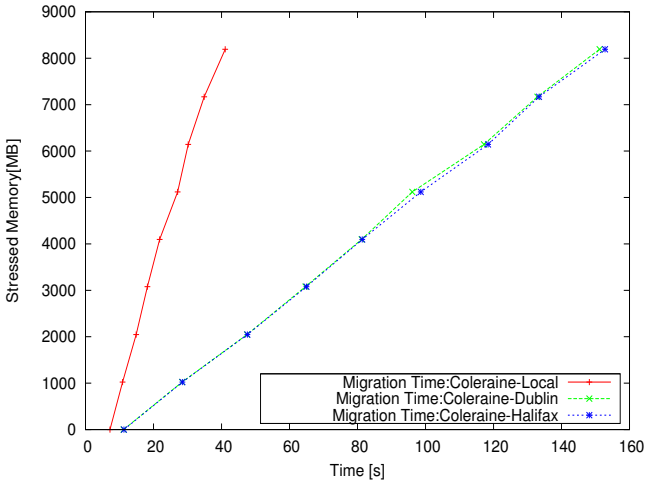

Fig. 7. Post-Copy LM –Stressed Memory with Time (s)

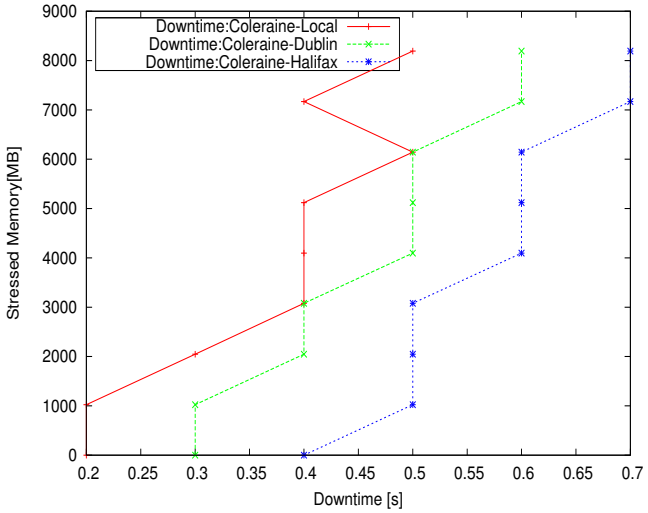

Fig. 8. Post-Copy Live Migration –Downtime(s)

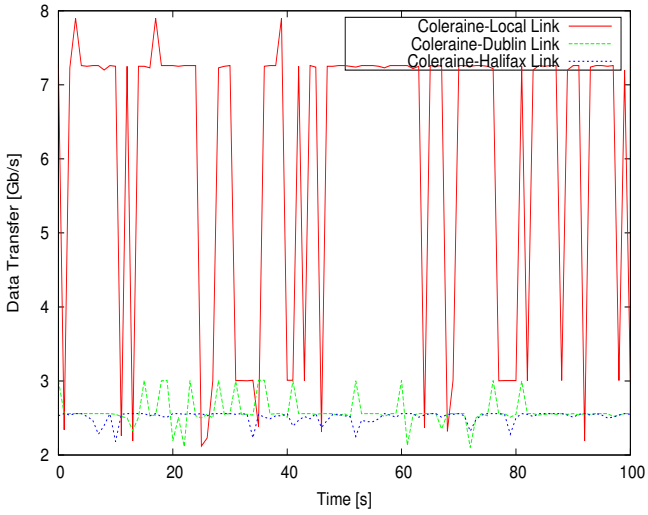

Fig. 9. Post-copy Live Migration- Bandwidth Utilisation

The downtime using the post-copy approach appears to be very stable and in the most of the cases lower than 700 ms as shown in TABLE III and Fig.8. The downtime varies from 200-700 ms as the amount of stressed memory increases. We observed significant differences in migration traffic behaviour for both Coleraine-Halifax and Coleraine-Dublin links while compared with the local 10Gb/s interface. As shown in Fig.9, local 10Gb/s interface sufficiently utilises the link up to the transfer capacity and the speed reaches maximal throughput of 8Gb/s. In contrast, we observed peak traffic up to 3Gb/s using the Coleraine-Dublin link and a pick of approximately 2.5Gb/s using the Coleraine-Halifax link. In all scenarios a VM with 2GB of the stressed memory was migrated using the post-copy algorithm that transferred approximately 4.7GB over the links.

VI. CONCLUSIONS

This paper examines the benefits of using high speed optical network infrastructures for migrations of live VMs. We have observed variation in LM duration for specific memory patterns considering different latency paths. In post-copy approach, the migration has taken tens of seconds for the local VMs with very intensive memory but with the same memory loads the LM took minutes to migrate in a higher latency path (e.g., live migration between Coleraine-Halifax VMs). However, our results show that the change in latency does not severely affect the downtime of the migrated VMs. We observed a stable and lower than 700 ms of downtime with a very little insignificant variation. High speed 10Gb/s interfaces with local servers also slightly extends possibilities of using standard pre-copy algorithm for more use cases and statistically decreases the risk of non convergence. Moreover, it has been shown that relatively large amounts of data (e.g., gigabytes of VM memory) need to migrate in order to fully utilise the 10Gb/s link. In future, we would like to develop new LM algorithms focusing on the relationship between the load generators used in this paper and the real-world applications in our high speed optical networks.

ACKNOWLEDGEMENTS

This research is supported by the InvestNI Digital Infrastructure Project and is funded by Invest Northern Ireland.

REFERENCES

[1] Resource Leasing and the Art of Suspending Virtual Machines, B.Sotomayor, R.Santiago Montero, I.Martín Llorente, I.Foster. The 11th IEEE International Conference on High Performance Computing and Communications (HPCC-09), June 25-27, 2009, Seoul, Korea.

[2] N. Bobroff, A. Kochut, and K. Beaty, BDynamic placement of virtual machines for managing SLA violations,[in Proc. IM, Munich, Germany, G2007, pp. 119–128.

[3] M. I. Biswas, G. Parr, S. McClean, P. Morrow and B. Scotney, "SLA-Based Scheduling of Applications for Geographically Secluded Clouds", 1st workshop on Smart Cloud Networks & Systems (SCNS'14), December 3-5 Paris, France, 2014.

[4] Welcome to Nova's developer documentation! http://docs.openstack.org/developer/nova/

[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. The 2nd conference on Symposium on Networked Systems Design & Implementation -Volume 2, pages 273–286, 2005.

[6] Ibrahim, K.Z., Hofmeyr, S., Iancu, C., Roman, E.: Optimized pre-copy live migration for memory intensive applications. In: Proc. 2011 Intl. Conf. High Performance Computing, Networking, Storage and Analysis, p. 40. ACM (2011)

[7] M. R. Hines and K. Gopalan. Post-copy based Live Virtual Machine Migration using Adaptive Pre-paging and Dynamic Self-ballooning. The 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, pages 51–60, 2009

[8] F. Moghaddam and M. Cheriet. Decreasing live virtual machine migration down-time using a memory page selection based on memory change PDF. Networking, Sensing and Control (ICNSC), 2010 International Conference on, pages 355 –359, April 2010.

[9] Aidan Shribman, Benoit Hudzia. "Pre-Copy and Post-Copy VM Live Migration for Memory Intensive Applications", LNCS 7640, pp. 539-547, Springer-Verlag Berlin Heidlberg 2013.

[10] Hibernia Networks: Security Through Diversity http://www.hiberniaatlantic.com/documents/LATENCYApril2007-1.pdf

[11] Stress(1): impose load on/stress test systems - Linux man page, http://linux.die.net/man/1/stress

[12] iPerf - The network bandwidth measurement tool, https://iperf.fr/